

**DATA SAMPLING METHOD AND APPARATUS WITH ALTERNATING
EDGE SAMPLING PHASE DETECTION FOR LOOP CHARACTERISTIC
STABILIZATION**

5 Cross-Reference to Related Application

 This application claims the benefit of U.S. Provisional Application No. 60/416,017, filed October 4, 2002, entitled "A 2.5–10GBPS CMOS TRANSCEIVER WITH ALTERNATING EDGE SAMPLING PHASE DETECTION FOR LOOP CHARACTERISTIC STABILIZATION."

10

Field of the Invention

 The invention pertains to clock and data recovery devices (CDRs), and to methods and circuitry for sampling data received over high speed links (e.g., high speed serial links).

15

Background of the Invention

 The term "receiver" is used herein in a broad sense to denote any device capable of receiving and decoding data that has been transmitted over a serial link or other link (and optionally also performing additional functions, which can include decrypting the received data and other operations related to decoding, reception, or decryption of the received data). For example, the term receiver can denote a transceiver that performs the functions of a transmitter as well as the functions of a receiver.

 The term "CDR" is used herein to denote a clock and data recovery device. Typically, a receiver includes a CDR. Throughout the disclosure, CDRs are described as devices that receive "data" and generate at least one sequence of samples of the data. It should be understood that the "data" can be a signal indicative of a clock, or a signal indicative of an arbitrary sequence of binary bits of data. Some CDRs include circuitry for recovering both a clock and data from one or more received signals. A "CDR," as the term is used herein (including in the claims) need not include circuitry for recovering both a clock and data from one or more received signals.

 The expression "phase detector" is used herein to denote a circuit that determines the phase error (ϕ_A) present between a sampling clock and data (being

sampled by the clock). A phase detector ("PD") is thus useful with a data sampling circuit including a feedback loop for controlling the phase of a sampling clock.

It should be understood that when phase error (ϕ_A), between a sampling clock and a data signal being sampled using the clock, is expressed herein in degrees, the quantity "360 degrees" corresponds to one bit time of the data signal (with "bit time" being synonymous with "bit period" and denoting the inverse of the bit rate) and typically does not correspond to one period of the sampling clock. Typically, a sampling clock employed in accordance with the invention (e.g., data sampling clock "*dCLK*" or edge sampling clock "*xCLK*" discussed below) has frequency equal to f/N , where " f " is the bit rate of the data signal being sampled and N is an integer greater than one (for example, $N = 4$ or N is another small integer). It should also be understood that when a clock is referred to herein as being " X " degrees out of phase with respect to another clock having the same frequency (e.g., when a raw edge sampling clock is said to be 180 degrees out of phase with a raw data sampling clock), the quantity "360 degrees" corresponds to one bit time of a data signal being sampled using the clocks, and typically does not correspond to one period of either one of the clocks. For example, in the below-described preferred implementation of the Fig. 3(a) system, raw data sampling clock "*idCLK*[0]" and raw edge sampling clock "*ixCLK*[0]" are referred to as being 180 degrees out of phase with respect to each other because clock "*ixCLK*[0]" lags clock "*idCLK*[0]" by one half of a bit time, and "*ixCLK*[0]" and raw edge sampling clock "*idCLK*[1]" are 180 degrees out of phase with respect to each other because clock "*idCLK*[1]" lags clock "*ixCLK*[0]" by one half of a bit time.

The expression "binary phase detector" is used herein to denote a circuit that determines whether the phase error (ϕ_A) present between a sampling clock and data (being sampled by the clock) is positive or negative, without determining the magnitude of the phase error.

A binary phase detector (binary PD) based on a 2x oversampling technique has been widely used in transceivers (for receiving data transmitted over high speed serial links) due to its simple structure. A transceiver including such a PD is described in the paper by A. Fiedler, et al., entitled "A 1.0625Gbps Transceiver with 2x-Oversampling and Transmit Signal Pre-Emphasis," ISSCC Digest of Technical Papers, pp. 238-239, Feb. 1997. However, the loop characteristics of a CDR using such a binary PD are dependent on the jitter amplitude of the input data stream and the CDR might not be robust over a wide range of jitter amplitude. An example of this uncertainty is

illustrated in Figure 1(a), which includes data eye diagrams, 2x based sampling clocks, edge distribution diagrams, and $I_{avg}-\phi_A$ curves for large and small amounts of jitter (where I_{avg} denotes average pumping current), assuming that some phase error (ϕ_A) is present between the sampling clocks and data. Typically, a CDR including a binary PD asserts a positive charge pump current (I_P) to cause advancement of the sampling clock phase when the sampling clock lags the data, and a negative charge pump current ($-I_P$) to cause retardation of the sampling clock phase when the sampling clock leads the data. Ideally (i.e., in the absence of jitter), the average pumping current (I_{avg}) is equal to the charge pump current (I_P) when the sampling clock lags the data and the average pumping current (I_{avg}) is equal to the negative charge pump current ($-I_P$) when the sampling clock leads the data. However, in a real world application in which there is high frequency jitter, the absolute value of the average pumping current (I_{avg}) is less than the absolute value of the charge pump current (when the sampling clock leads or lags the data) since the binary PD makes an up/down decision (a decision as to whether to increase or decrease the sampling clock phase) per each data transition and I_{avg} is proportional to an average difference of the up/down decisions (represented by the shaded area in Figure 1(a)). Therefore, for the same charge pump current, larger jitter reduces the absolute magnitude of the average pumping current and reduces the gain K_{PD} (the slope of the I_{avg} versus ϕ_A curve). So, the gain K_{PD} and the absolute magnitude of I_{avg} are inversely proportional to the amount of jitter, as indicated by the bottom graph in Fig. 1(a).

It should be understood that the expression “average” pumping current (I_{avg}) herein denotes an average of instantaneous values of the charge pump current over a time interval in which averaged phase error (averaged ϕ_A) is positive or negative; not a time interval over which averaged phase error (averaged ϕ_A) is zero. An average of instantaneous values of charge pump current, over a time interval in which averaged phase error (averaged ϕ_A) is zero, would be zero (or nearly zero) and would typically be neither proportional nor inversely proportional to the amount of jitter.

The unstable gain (K_{PD}) due to jitter variation, of a conventional CDR that includes a binary PD based on a 2x oversampling technique as described with reference to Fig. 1(a), results in unreliable CDR loop characteristics.

Conventional CDRs that include 2x oversampling PDs do not support any method to measure jitter. Estimation of jitter can be accomplished in accordance with

the invention using a modified version of the “dead zone” PD described in Y. Moon, et al., “A 0.6 – 2.5Gbaud CMOS Tracked 3x Oversampling Transceiver with Dead Zone Phase Detection for Robust Clock/Data Recovery,” ISSCC Digest of Technical Papers, pp. 212-213, Feb. 2001, and in S. Lee, et al., “A 5Gb/s 0.25 μ m CMOS Jitter-Tolerant Variable-Interval Oversampling Clock/Data Recovery Circuit,” ISSCC Digest of Technical Papers, pp. 256-257, Feb. 2002. Because the dead zone PD described in the cited papers by Moon, et al. and Lee, et al. is based on 3x oversampling, a CDR including the dead zone PD must employ 3x oversampling and thus requires more hardware to implement and consumes more power than does a CDR that employs 2x oversampling. The noted problems of the prior art can be overcome in accordance with the invention by using a CDR that employs 2x oversampling, has reliable CDR loop characteristics, and includes an embodiment of the inventive alternating edge sampling phase detector (“AES PD”) which establishes a dead zone.

Summary of the Invention

In a class of embodiments, the invention is a clock and data recovery device (CDR) that includes a binary phase detector (binary PD), employs a 2x oversampling technique, and is not subject to unstable gain (K_{PD}) due to jitter variation. The inventive CDR includes a data loop configured to generate at least one sampling clock in response to a charge pump current I_P having an average value I_{avg} , and is configured to generate the charge pump current I_P in response to feedback indicative of phase error (ϕ_A) between the sampling clock and the data being sampled and also indicative of the amount of the jitter of the data being sampled. The relation between I_{avg} and ϕ_A is typically such that the “ I_{avg} versus ϕ_A ” curve has a linear portion (at small values of ϕ_A) and the data loop gain K_{PD} is the slope of this linear portion. A constant gain K_{PD} (independent of the amount of the data’s jitter) is achieved in accordance with the invention by adjusting the charge pump current I_P to track amount of the jitter. Preferably, the CDR includes a jitter estimating circuit and is configured to make the absolute value of I_P proportional to the amount of the jitter, thereby causing the average pumping current (I_{avg}) and data loop gain (K_{PD}) to be at least substantially independent of the amount of the jitter.

In preferred embodiments, the inventive CDR is an alternating edge sampling CDR that includes a frequency acquisition loop, a data loop, and a dead zone width

control loop. At power-on, the frequency acquisition loop is activated to set the CDR operation frequency to match (e.g., within the range of ± 200 ppm) an external reference clock frequency. After frequency lock is obtained, the data loop and dead zone width control loop are activated.

5 In some embodiments, the data loop includes sampling circuitry configured to generate data samples by sampling an input data signal (typically indicative of a sequence of input data bits) using a data sampling clock whose frequency is equal to the input data rate (where "input data rate" denotes the rate of occurrence of the "one" and "zero" bits indicated by the input data signal). In other embodiments, the data loop
10 includes sampling circuitry configured to generate N sequences of data samples (where N is an integer greater than one) using a set of N data sampling clocks, all having the same frequency but each having a different phase than each of the other data sampling clocks. In the latter embodiments, each sequence of data samples is generated by sampling the input data signal using a different one of the data sampling clocks, and the
15 frequency of each data sampling clock is equal to R/N , where R is the input data rate, and "input data rate" denotes the rate of occurrence of the "one" and "zero" bits indicated by the input data signal.

 In each embodiment mentioned in the previous paragraph, the data loop preferably generates a raw data sampling clock, a raw edge sampling clock, and an
20 edge sampling clock (for each data sampling clock), and generates the relevant data sampling clock by applying a delay to the raw data sampling clock (a fixed delay in the case that the input data signal has constant jitter), adjusts the phase of the data sampling clock to the center of the data eye, and keeps the raw edge sampling clock 180 degrees out of phase with the raw data sampling clock. Preferably, the data loop includes
25 circuitry (sometimes referred to herein as an alternating edge sampling phase detector or "AES PD") which generates the edge sampling clock by applying a sequence of small positive and negative offsets to the raw edge sampling clock (a positive offset is applied when the AES PD is in an "UP" state, and a negative offset is applied when the AES PD is in a "DOWN" state), in the sense that in the UP state it delays the raw edge
30 sampling clock by slightly less than it delays the raw data sampling clock (to generate the data sampling clock) and in the DOWN state it delays the raw edge sampling clock by slightly more than it delays the raw data sampling clock (to generate the data sampling clock). Preferably, the data loop also generates clock phase control signals (feedback signals) in response to samples of the input data signal generated using the

edge sampling clock and the data sampling clock. The phase difference between the edge sampling clock (with the positive offset) and the edge sampling clock (with the negative offset) is known as a “dead zone.” The data loop controls the phase of the data sampling clock in response to the clock phase control signals. In typical embodiments, the data loop includes a voltage controlled oscillator (VCO) which generates the raw edge sampling clock and a raw data sampling clock, and charge pump circuitry which generates charge pump current I_P (and a voltage, proportional to the charge pump current, for controlling the VCO) in response to the clock phase control signals.

In preferred embodiments, the data loop generates a raw data sampling clock (sometimes referred to herein as $idCLK$) and a raw edge sampling clock (sometimes referred to herein as $ixCLK$), generates a data sampling clock (sometimes referred to herein as $dCLK$) in response to the raw data sampling clock, adjusts the phase of the data sampling clock to the center of the data eye, keeps the raw edge sampling clock 180 degrees out of phase with the raw data sampling clock, and generates an edge sampling clock (sometimes referred to as $xCLK$) by applying a sequence of small positive and negative offsets to the raw edge sampling clock. Typically, the positive offset is applied to advance the phase of $xCLK$, relative to that of $dCLK$, when a control bit (e.g., “up_det” asserted by finite state machine 4 of Fig. 3(a)) indicates that the AES PD is in an “UP” state, and the negative offset is applied to retard the phase of $xCLK$ relative to that of $dCLK$ when the control bit indicates that the AES PD is in a “DOWN” state. Preferably, the data loop generates charge pump control bits (e.g., the bits “up[0:3]” and “down[0:3]” of Fig. 3(a)) in response to samples of the input data signal generated using the data sampling clock and the edge sampling clock, and charge pump and VCO circuitry of the data loop controls the phase of the raw data sampling and raw edge sampling clocks in response to the charge pump control bits.

The dead zone width control loop includes a jitter estimating circuit configured to generate a jitter signal indicative of the input data’s jitter. In response to the jitter signal, the dead zone width control loop adjusts the width (W_D) of the dead zone defined by the edge sampling clock (e.g., controls the magnitude of the positive and negative offsets that the data loop applies to the raw edge sampling clock to generate the edge sampling clock), and preferably also controls the charge pump current (I_P). The dead zone width (W_D) is automatically adjusted during operation (in response to the jitter signal) to cause the edges of the dead zone to track the boundaries of the data eye, for example as indicated in Figure 1(b). In Fig. 1(b), the data sampling clock $dCLK$

has phase error ϕ_{Δ} (relative to the center of the data eye), two versions of the edge sampling clock $xCLK$ are shown (an “up” version having phase $(\phi - 180 + W_D/2)$ degrees, and a “down” version having phase $(\phi - 180 - W_D/2)$ degrees, where ϕ is the phase of $dCLK$), and the dead zone width W_D is varied to be proportional to the amount of the data’s jitter. In preferred embodiments, the charge pump current (I_P) is controlled in response to the jitter signal to cause the average pumping current (I_{avg}) and data loop gain (K_{PD}) to be at least substantially independent of amount of the jitter and the dead zone width.

In preferred embodiments, the jitter estimating circuit of the dead zone width control loop estimates input data jitter by counting (in the digital domain) the number of times that the data loop changes the phase of the raw data sampling clock (e.g., the number of charge pump control bits asserted with magnitude “up = 1” or “down = 1” by finite state machine 4 of Fig. 3(a)) during a predetermined number of valid input data transitions. The width of the dead zone is increased (or decreased) as this count increases (or decreases). To prevent interaction between the data loop and the dead zone width control loop, the bandwidth of the dead zone width control loop is preferably much smaller than that of the data loop.

When the raw edge sampling clock of preferred embodiments of the invention is locked to the data transition edge, the edge sampling clock samples data with positive or negative offset from the data transition edge. In these embodiments, the AES PD of the invention has two sampling states: an UP state and a DOWN state. In the UP state, the edge sampling clock samples data with positive offset with respect to the data transition edge (i.e., at the left boundary of the dead zone, as shown in Figs. 1(b) and 4(a)) and the data sampling clock also samples the data. In the DOWN state, the edge sampling clock samples data with negative offset with respect to the data transition edge (i.e., at the right boundary of the dead zone) and the data sampling clock also samples the data. When locked to the input data, the AES PD of the invention undergoes a transition between the UP and DOWN state at every data transition. When not locked to the input data, the AES PD remains in a first one of the UP state and the DOWN state while the data loop adjusts the phase of the raw data sampling clock (and the raw edge sampling clock) to reduce the phase difference between the data sampling clock and the center of the data eye.

In preferred embodiments, the data loop adjusts the phase of the raw data sampling clock by asserting charge pump control signals (e.g., charge pump control bits

“up[0:3]” and “down[0:3]” bits of Fig. 3(a)) indicative of the offset of the data sampling clock from the center of the data eye to charge pump circuitry that controls a VCO which generates the data sampling clock (e.g., charge pump circuitry 12 of Fig. 3(a) which controls VCO 20). In response, the current I_P generated by the charge pump circuitry causes the VCO to advance (or retard) the phase of the raw data sampling clock when the charge pump control signals indicate that the phase of the data sampling clock lags (or leads) the data eye. The charge pump control signals are asserted to the jitter estimating circuit of the dead zone width control loop as well as to the charge pump circuitry of the data loop. For example, the charge pump control bits “up[0:3]” and “down[0:3]” bits of Fig. 3(a), which consist of bits up_i and $down_i$ (where $i = 0, 1, 2$, or 3, and “ i ” is indicative of one of the below-mentioned layers), are asserted to charge pump circuitry 12 and at least some of these bits are asserted to dead zone width control circuit 14. A stream of the up_i and $down_i$ bits includes more $up_i = 1$ and $down_i = 1$ bits (per a predetermined number of data transitions) when the input data exhibits more jitter, and fewer $up_i = 1$ and $down_i = 1$ bits when the input data exhibits less jitter. In response to the charge pump control signals, the dead zone width control loop increases the dead zone width when the input data exhibits more jitter. When the dead zone widens, the data loop asserts fewer $up_i = 1$ and $down_i = 1$ bits (assuming that the jitter does not change) per the predetermined number of data transitions.

Other aspects of the invention are a binary phase detector (PD) for use in an embodiment of the inventive CDR, data loop circuitry for use in an embodiment of the inventive CDR, dead zone width control loop circuitry for use in an embodiment of the inventive CDR, and a jitter estimating circuit for use in a dead zone width control loop.

Another aspect of the invention is a method for sampling data having jitter, including the steps of: (a) generating at least one sampling clock in response to a charge pump current, where at least one said sampling clock is a data sampling clock and the charge pump current determines the phase of the data sampling clock; (b) generating data samples by sampling the data using the data sampling clock; and (c) generating the charge pump current in response to feedback, where the feedback is indicative of phase error between the data sampling clock and the data, the feedback is also indicative of the jitter, the charge pump current has an average current value that is at least substantially independent of the amount of the jitter over each time interval over which ϕ_{av} is nonzero, where ϕ_{av} is an average of instantaneous values of said phase error over the time interval, and the average current value is an average of instantaneous values of

the charge pump current over said time interval. Typically, the charge pump current is a positive current (I_P) when the phase error is negative, the charge pump current is a negative current, $-(I_P)$, when the phase error is positive, and step (c) includes the step of controlling the absolute value of the charge pump current to cause said absolute value
5 be proportional to the jitter. Preferably, another sampling clock generated during step (a) is a first clock whose phase is determined by the charge pump current, and step (c) includes the steps of: modulating the first clock to generate an edge sampling clock such that the edge sampling clock defines a dead zone having a dead zone width; generating additional samples of the data in response to the edge sampling clock; and
10 generating the feedback in response to the data samples and the additional data samples. In some embodiments, the data determines a data eye, and step (c) includes the step of automatically adjusting the dead zone width to cause edges of the dead zone to track boundaries of the data eye.

15 Brief Description of the Drawings

Fig. 1(a) includes data eye diagrams, edge distribution diagrams (showing conventional 2x based sampling clocks), and $I_{avg}-\phi_{\Delta}$ curves for large and small amounts of jitter (where I_{avg} denotes average pumping current), assuming that phase error (ϕ_{Δ}) is present between the sampling clocks and data.

20 Fig. 1(b) includes data eye diagrams, edge distribution diagrams (showing the inventive, phase-modulated 2x based sampling clocks dCLK and xCLK), and $I_{avg}-\phi_{\Delta}$ curves for large and small amounts of jitter, assuming that phase error (ϕ_{Δ}) is present between the sampling clocks and data.

Fig. 2(a) is a graph showing three jitter transfer curves (each assuming a
25 different amount of jitter) for a conventional CDR including a binary PD and based on a 2x oversampling technique.

Fig. 2(b) is a graph showing three jitter transfer curves (each assuming one of the jitter amounts assumed in a corresponding curve in Fig. 2(a)) for an embodiment of the inventive CDR (which includes a binary PD and is based on a 2x oversampling
30 technique).

Fig. 3(a) is a block diagram of a CDR that embodies the invention and includes an alternating edge-sampling binary PD.

Fig. 3(b) is a block diagram of a dead zone generator for use in the Figure 3(a) circuit.

Fig. 4(a) is a timing diagram of signals asserted during operation of the Fig. 3(a) circuit.

5 Fig. 4(b) is a state diagram of the Fig. 3(a) circuit.

Fig. 5 is a block diagram of a transceiver that embodies the invention, in which alternating edge-sampling CDR 33 is implemented as shown in Fig. 3(a).

Fig. 6(a) shows an edge sampling clock (of a conventional binary PD) and the input data transition probability density function (PDF) in a lock state of the binary PD, assuming no phase error (ϕ_A) between the data sampling clock (not shown) and data.

Fig. 6(b) shows an edge sampling clock of the binary PD of Fig. 6(a) and the input data transition PDF of such binary PD, assuming nonzero phase error between the data sampling clock (not shown) and data.

Fig. 7(a) shows an edge sampling clock (of a dead zone PD) and the input data transition PDF in a lock state of the dead zone PD, assuming no phase error (ϕ_A) between the data sampling clock (not shown) and data.

Fig. 7(b) shows an edge sampling clock of the dead zone PD of Fig. 7(a) and the input data transition PDF of such dead zone PD, assuming nonzero phase error between the data sampling clock (not shown) and data.

20 Fig. 8 shows an edge sampling clock of an embodiment of the inventive PD, and the input data transition PDF of such PD, assuming nonzero phase error (ϕ_A) between the data sampling clock (not shown) and data being sampled.

Fig. 9(a) is a block diagram of some elements of a transceiver that includes an embodiment of the inventive CDR, with simplified schematic diagrams of preferred implementations of VCOs 20. PLL 38 of Fig. 9(a) (identified as "TXPLL") is included in the transmitter portion of the transceiver, and PLL 39 (identified as "RXPLL") is included in the CDR portion of the transceiver.

Fig. 9(b) is a single stage delay cell of VCO 20 of RXPLL 39 of Fig. 9(a).

Fig. 10 is a schematic diagram of a cascaded sampler employed in preferred
30 embodiments of the invention.

Fig. 11(a) is a schematic diagram of a conventional charge pump, and a waveform indicating the charge pump current I_p that would be generated by each charge pump of charge pump circuitry 12 of Fig. 3(a), in response to a specific

sequence of up_i and $down_i$ bits, if such charge pump were implemented as the conventional charge pump of Fig. 11(a).

Fig. 11(b) is a schematic diagram of a conventional charge pump, and a waveform indicating the charge pump current I_p that would be generated by each charge pump of charge pump circuitry 12 of Fig. 3(a), in response to a specific sequence of up_i and $down_i$ bits, if such charge pump were implemented as the conventional charge pump of Fig. 11(b).

Fig. 11(c) is a schematic diagram of a charge pump designed in accordance with the present invention, and a waveform indicating the charge pump current I_p that would be generated by each charge pump of charge pump circuitry 12 of Fig. 3(a), in response to a specific sequence of up_i and $down_i$ bits, if such charge pump were implemented as shown in Fig. 11(c).

Fig. 12(a) is a block diagram of a dead zone width control loop that can be used to implement the dead zone width control loop of Fig. 3(a).

Fig. 12(b) is the state diagram of the FSM of Fig. 12(a).

Fig. 13 is a block diagram of a frequency lock detector that can be used to implement the frequency lock detector block of Fig. 3(a).

Detailed description of the Invention

We sometimes refer to a circuit herein as a PD (e.g., an “alternating edge sampling PD”) when it includes not only a circuit that determines the phase error (ϕ_A) present between a sampling clock and data (being sampled by the clock), but also additional circuitry (e.g., data sampling circuitry). For example, alternating edge sampling phase detector (“AES PD”) 2 of Fig. 3(a) includes data sampling circuitry (e.g., circuit 8) as well as circuitry for generating signals (“up[0:3]” and “down[0:3]”) indicative of whether the data sampling clock employed by the data sampling circuitry leads or lags the data being sampled.

In a class of embodiments, the invention is a clock and data recovery device (CDR) that includes a binary phase detector (binary PD), employs a 2x oversampling technique, and is not subject to unstable gain (K_{PD}) due to jitter variation. The inventive CDR includes a data loop configured to generate sampling clocks in response to a charge pump current I_p having an average value I_{avg} , and is configured to generate the charge pump current I_p in response to feedback indicative of phase error (ϕ_A) between the sampling clocks and the data being sampled and also indicative of jitter of the data

being sampled. The relation between I_{avg} and ϕ_{Δ} is typically such that the “ I_{avg} versus ϕ_{Δ} ” curve has a linear portion (at small values of ϕ_{Δ}) and the gain K_{PD} is the slope of this linear portion. A constant K_{PD} (independent of the data’s jitter) is achieved in accordance with the invention by adjusting the charge pump current I_P to track the data’s jitter. Preferably, the CDR includes a jitter estimating circuit and is configured to make the absolute value of I_P proportional to the jitter, thereby causing the average pumping current (I_{avg}) and gain (K_{PD}) to be at least substantially independent of the jitter.

As elsewhere herein, the expression “average” pumping current (I_{avg}) denotes an average of instantaneous values of the charge pump current over a time interval in which averaged phase error, ϕ_{av} (where “ ϕ_{av} ” is the average of instantaneous values of phase error ϕ_{Δ} over the time interval), is nonzero; not a time interval over which averaged phase error (ϕ_{av}) is zero. An average of instantaneous values of charge pump current, over a time interval in which averaged phase error (ϕ_{Δ}) is zero, would always be zero (or nearly zero) and would typically be neither proportional nor inversely proportional to the amount of jitter.

An embodiment of the inventive CDR and a conventional CDR have been analyzed using a time-step simulator. A randomly generated data stream was fed to the conventional CDR (which included a conventional binary PD and was based on a 2x oversampling technique) and random and deterministic jitter (RJ and DJ) were added to the data stream. The resulting jitter transfer curves are plotted for some jitter corners in Figure 2(a). A randomly generated data stream was also fed to the inventive CDR (which included an AES binary PD) and random and deterministic jitter (RJ and DJ) having the same characteristics were added to the data stream. The resulting jitter transfer curves are plotted for some jitter corners in Figure 2(b). As shown in Figure 2(b), the inventive CDR has a substantially constant transfer curve (independent of the amount of jitter). As shown in Figure 2(a), the conventional CDR has very different transfer curves depending on the amount of jitter.

The details of a preferred embodiment of the invention will next be discussed below with reference to Figs. 3(a), 3(b), 4(a), 4(b), 12(a), 12(b), and 13.

Figure 3(a) is a block diagram of a CDR which includes an alternating edge sampling binary PD (“AES PD” 2). The CDR circuitry shown in Fig. 3(a) is composed of four circuits (to be referred to as “layers”), each coupled to receive the same

differential input signal (RDP, RDN). Typically, the four layers operate in parallel and each layer operates at a different phase in the following sense. The first layer generates an adjusted data sampling clock ($dCLK$) in response to a first data sampling clock ($idCLK[0]$) from VCO 20, generates an adjusted edge sampling clock ($xCLK$) in response to a first edge sampling clock ($ixCLK[0]$) from VCO 20, and generates a stream of data samples DS[0] of the differential input signal using the first adjusted data sampling clock and the first adjusted edge sampling clock. The second layer generates a second adjusted data sampling clock ($dCLK[1]$) in response to a second data sampling clock, $idCLK[1]$ (which is delayed by one bit time relative to the first data sampling clock) from VCO 20, generates a second adjusted edge sampling clock ($xCLK[1]$) in response to a second edge sampling clock, $ixCLK[1]$ (which is delayed by one bit time relative to the first edge sampling clock) from VCO 20, and generates a stream of data samples DS[1] of the differential input signal using the second adjusted data sampling clock and the second adjusted edge sampling clock. The third layer generates a third adjusted data sampling clock ($dCLK[2]$) in response to a third data sampling clock, $idCLK[2]$ (which is delayed by two bit times relative to the first data sampling clock) from VCO 20, generates a third adjusted edge sampling clock ($xCLK[2]$) in response to a third edge sampling clock, $ixCLK[2]$ (which is delayed by two bit times relative to the first edge sampling clock) from VCO 20, and generates a stream of data samples DS[2] of the differential input signal using the third adjusted data sampling clock and the third adjusted edge sampling clock. The fourth layer generates a fourth adjusted data sampling clock ($dCLK[3]$) in response to a fourth data sampling clock, $idCLK[3]$ (which is delayed by three bit times relative to the first data sampling clock) from VCO 20, generates a fourth adjusted edge sampling clock ($xCLK[3]$) in response to a fourth edge sampling clock, $ixCLK[3]$ (which is delayed by three bit times relative to the first edge sampling clock) from VCO 20, and generates a stream of data samples DS[3] of the differential input signal using the fourth adjusted data sampling clock and the fourth adjusted edge sampling clock. The four data sample streams, denoted as "DS[0:3]" in Fig. 3(a), have a combined sample rate that is four times the sample rate of each individual stream (DS[0], DS[1], DS[2], or DS[3]).

One of the major merits of the invention is that it can easily be implemented with a parallel architecture (as is the Fig. 3(a) circuit). If a system that embodies the invention is implemented with a parallel architecture, it can operate at lower speed than if it were not implemented with a parallel architecture. For example, the preferred

implementation of the Fig. 3(a) circuit generates data samples DS[0:3] with the same sample rate that could be attained by omitting three of its four layers and operating the remaining layer in response to data sampling and edge sampling clocks having frequency four times greater than the frequency (equal to 2.5 GHz, in one embodiment) of each of data sampling and edge sampling clocks *idCLK*[0:3] and *ixCLK*[0:3] of Fig. 3(a). In such a preferred implementation of the Fig. 3(a) circuit, for each value of *i* (where *i* = 0, 1, 2, or 3), raw data sampling clock “*idCLK*[*i*]” and raw edge sampling clock “*ixCLK*[*i*]” are 180 degrees out of phase with respect to each other (in the sense that clock “*ixCLK*[0]” lags clock “*idCLK*[0]” by one half of a bit time. Similarly, for each value of *i* in the range $0 \leq i \leq 2$, “*ixCLK*[*i*]” and “*idCLK*[*i*+1]” are 180 degrees out of phase with respect to each other in the sense that “*idCLK*[*i*+1]” lags clock “*ixCLK*[*i*]” by one half of a bit time, and “*ixCLK*[3]” and “*idCLK*[0]” are 180 degrees out of phase with respect to each other in the sense that “*idCLK*[0]” lags clock “*ixCLK*[3]” by one half of a bit time.

The four layers of Fig. 3(a) are identical to each other. Circuit 2a of AES phase detector 2 belongs to the first layer, circuit 2b of AES phase detector 2 belongs to the second layer, circuit 2c of AES phase detector 2 belongs to the third layer, and circuit 2d of AES phase detector 2 belongs to the fourth layer. In typical operation, each of circuits 2a, 2b, 2c, and 2d, is coupled to receive the same differential input signal (RDP, RDN).

In variations on the Fig. 3(a) embodiment, the AES phase detector of the inventive CDR circuitry has a single layer that is coupled to receive one differential input signal (e.g., three of circuits 2a, 2b, 2c, and 2d are omitted), so that only a single raw data sampling clock (*idCLK*) is generated (and employed) and a single raw edge sampling clock (*ixCLK*) is generated and employed. In some such variations, the frequency of the raw data sampling clock (*idCLK*) and the frequency of the raw edge sampling clock (*ixCLK*) matches the bit rate.

With reference again to Fig. 3(a), AES phase detector 2 includes four identical data sampling circuits 6, four identical data sampling circuits 8, four identical finite state machines 4, and four identical dead zone generation circuits 10. Each of circuits 2a, 2b, 2c, and 2d includes one of the data sampling circuits 6, one of the data sampling circuits 8, one finite state machine (“PD FSM”) 4, and one dead zone generation circuit 10, connected as shown. Some other elements of Fig. 3(a), such as voltage controlled oscillator (“VCO”) 20 and charge pump circuitry 12, are shared by all four layers but

generate separate output signals for each layer. For example, VCO 20 generates a different raw data sampling clock (*idCLK*[0], *idCLK*[1], *idCLK*[2], and *idCLK*[3]) for each layer and a different raw edge sampling clock (*ixCLK*[0], *ixCLK*[1], *ixCLK*[2], and *ixCLK*[3]) for each layer. For another example, circuitry 12 preferably includes
5 four identical charge pump circuits, each for sourcing or sinking a current I_p (to or from the capacitors of filter 15) to cause filter 15 to assert a low-pass-filtered voltage to VCO 20 to control VCO 20's generation of the raw data sampling clock and raw edge sampling clock for a different one of the layers.

Some other elements of Fig. 3(a) are shared by all four layers. Examples of such
10 elements are dead zone width control circuit 14, switch 19, and frequency acquisition circuitry 13 (which generates a single control signal for switch 19 that is shared by all layers).

The Fig. 3(a) system includes a frequency acquisition loop (comprising frequency acquisition circuitry 13 and VCO 20) which is shared by all four of the
15 layers of Fig. 3(a). Each of the four layers of Fig. 3(a) also comprises two loops: a data loop (including one set of circuits 4, 6, 8, and 10, one layer of charge pump circuitry 12, and one layer of VCO 20), and a dead zone width control loop (including circuit 14 and one set of circuits 4, 6, 8, and 10).

Circuits 12, 14, 15, and 19 are clock control circuitry in the sense that they
20 control generation of clocks by VCO 20 and dead zone generation circuit 10. Other embodiments of the invention include other implementations of clock control circuitry.

After power-on, the frequency acquisition loop is initially activated to set the CDR operation frequency within the range of ± 200 ppm apart from the frequency of an external reference clock ("Ref_CLK"). After frequency lock is obtained, the frequency
25 acquisition loop is deactivated and the data loops and dead zone width control loops are activated.

Each data loop provides a data sampling clock (*idCLK*[0], *idCLK*[1], *idCLK*[2], or *idCLK*[3]) and an edge sampling clock (*ixCLK*[0], *ixCLK*[1], *ixCLK*[2], or *ixCLK*[3]) to a different one of the four circuits 10 of AES phase detector 2, and each
30 circuit 10 generates an adjusted data sampling clock (*dCLK*[0], *dCLK*[1], *dCLK*[2], or *dCLK*[3]) in response to the data sampling clock that it receives (and in response to control signals generated by one of finite state machines 4) and generates an adjusted edge sampling clock (*xCLK*[0], *xCLK*[1], *xCLK*[2], or *xCLK*[3]) in response to the edge sampling clock that it receives (and in response to control signals generated by one of

finite state machines 4). Each sampling circuit 8 uses the adjusted data sampling clock that it receives to produce a stream of data samples (DS[0], DS[1], DS[2], or DS[3]) in response to the differential input signal, RDP, RDN. Each data loop adjusts the phase of the data sampling clock that it generates to the center of the data eye and keeps the edge sampling clock that it generates 180 degrees out of phase with the corresponding data sampling clock.

The dead zone width control loop adjusts the edge sampling clocks ($ixCLK[0]$, $ixCLK[1]$, $ixCLK[2]$, and $ixCLK[3]$) in response to a set of dead zone width control bits (" $ctr[2:0]$ ") generated by dead zone width control circuit 14, to track the boundary of the data eye. Circuit 14 preferably generates the dead zone width control bits as follows. Circuit 14 receives a sequence of bits up_i (a sequence of bits up_1 , up_2 , up_3 , or up_4 from one layer of AES PD 2), and a sequence of bits $down_i$ (a sequence of bits $down_1$, $down_2$, $down_3$, or $down_4$ from the same layer of AES PD 2). Each bit up_i is a bit of a word $up[0:3]$ generated by AES PD 2, and each bit $down_i$ is a bit of a word $down[0:3]$ generated by AES PD 2. Circuit 14 also receives a sequence of bits v_i (a sequence of bits v_1 , v_2 , v_3 , or v_4 from one layer of AES PD 2), where each bit v_i is a bit of a word $v[0:3]$ from AES PD 2 that indicates whether there has been a transition (during the last two cycles of clock $dCLK$) of the input data signal RDP, RDN received by the relevant layer. Circuit 14 counts (in the digital domain, and for any one index value " i ") the number of " up_i " and " $down_i$ " bits that satisfy " $up_i = 1$ " or " $down_i = 1$ " and that occur during a predetermined number of valid data transitions (as indicated by the bits v_i of a sequence of the words $v[0:3]$, for any one index value " i "), in a manner to be described with reference to Figs. 12(a) and 12(b). To prevent the interaction between each data loop and the corresponding dead zone width control loop, the bandwidth of the dead zone width control loop is much smaller than that of the data loop.

VCO 20 asserts four raw edge sampling clock signals $ixCLK[0:3]$ (one for each layer) to AES PD 2 and one of the raw edge sampling clock signals ($ixCLK[0]$, $ixCLK[1]$, $ixCLK[2]$, or $ixCLK[3]$) to frequency acquisition circuitry 13, and also asserts four raw data sampling signals $idCLK[0:3]$ (one for each layer) to AES PD 2. Frequency acquisition circuitry 13 includes frequency divider 17, phase detector and charge pump circuit 18, and frequency lock detection circuit 16, connected as shown. To obtain frequency lock with external reference clock "Ref_CLK," lock detection circuit 16 asserts to switch 19 a control signal (indicating that frequency lock has not

yet been obtained) to cause switch 19 to feed back control voltages from circuit 18 (which are low pass filtered by filter 15) to VCO 20. When the PLL implemented by elements 17, 18, 15, and 20 has caused VCO 20 to lock the signals $ixCLK[0:3]$ and $idCLK[0:3]$ to the reference clock, lock detection circuit 16 asserts to switch 19 a
 5 control signal (indicating that frequency lock has been obtained) causing switch 19 to connect the outputs of charge pump circuitry 12 to VCO 20.

Fig. 13 is a block diagram of a frequency lock detector that can be used to implement lock detection circuit 16 of Fig. 3(a). A reference time interval ($16k \times T_{ref}$), where $1/T_{ref}$ is the frequency of the external reference clock Ref_CLK) is generated
 10 from frequency divider block 80 of the Fig. 13 circuit. The “VCO” signal asserted to sense-amp type flip-flop (SAFF) 81 and sixteen-bit counter 83 is a divided-down version (asserted by divider 17 of Fig. 3(a)) of one of the signals $ixCLK$ output by VCO 20. Because the VCO and Ref_CLK signals comprise an asynchronous clock boundary, sense-amp type flip-flop (SAFF 81) is used to avoid a metastable situation.
 15 By using frequency divider 80 and SAFF 81, the frequency lock detector of Fig. 13 has less hardware complexity than does the frequency lock detection circuitry disclosed in the above-cited paper by Y. Moon, et al. The Fig. 13 detector also includes rising edge detection circuit 82, 16-bit counter 83, and decision logic 84, connected as shown. Decision logic 84 of Fig. 13 applies two different thresholds for getting into (250 ppm)
 20 or leaving (1000 ppm) the lock state for robust operation. The control signal “F lock” output from logic 84 is asserted to switch 19.

As noted, charge pump circuitry 12 of Fig. 3(a) preferably includes four identical charge pump circuits. Each charge pump circuit is for sourcing or sinking a current I_p to or from the capacitors of low pass filter 15. All the charge pump circuits
 25 drive a single low pass filter 15. The voltage across the capacitors of filter 15 is asserted to VCO 20 to control clock signal generation by VCO 20. Each of the four charge pump circuits is preferably implemented as shown in Fig. 11(c). Node N of Fig. 11(c) can be coupled (via switch 19) to Node M of filter 15. When node N of Fig. 11(c) is coupled to Node M of filter 15, the capacitors of filter 15 are charged (or discharged)
 30 in response to the current I_p that is sourced to (or sunk from) the capacitors by the Fig. 11(c) circuit. When switch 19 couples the relevant charge pump circuit (within circuitry 12) to VCO 20 and filter 15, filter 15 low-pass filters the voltage V_C across the capacitors (the voltage at node M of filter 15), and the low-pass-filtered voltage V_C is asserted to VCO 20.

The Fig. 11(c) circuit is coupled to receive a pair of charge pump control bits “ up_i ” and “ $down_i$ ” from circuit 2a, 2b, 2c, or 2d (of AES PD 2) and to receive a pair of bias potentials V_p and V_n . The bit “ up_i ” is denoted as “ up ” in Fig. 11(c). The Fig. 11(c) circuit generates the complement (denoted as “ $downb$ ” in Fig. 11(c)) of the bit “ $down_i$ ”.

- 5 In response to $up_i = 1$ and $down_i = 0$ (when the Fig. 11(c) circuit is coupled to filter 15), as can occur when the raw data clock $idCLK$ lags the input data, the Fig. 11(c) circuit asserts a positive current I_p to the capacitors of filter 15, to charge these capacitors. In response to the bits $up_i = 0$ and $down_i = 1$ (when the Fig. 11(c) circuit is coupled to filter 15), as can occur when the raw data clock $idCLK$ leads the input data, the Fig.
- 10 11(c) circuit sinks the current I_p from the capacitors of filter 15, to discharge these capacitors. In response to the bits $up_i = 0$ and $down_i = 0$ (when the Fig. 11(c) circuit is coupled to filter 15), as can occur when the raw data clock $idCLK$ is locked with the input data, no significant current flows to or from the capacitors of filter 15.

- Still with reference to Fig. 11(c), bias potentials V_p and V_n determine the
- 15 absolute value of the current I_p as follows, when PMOS transistor P1 of Fig. 11(c) is implemented as a multi-finger transistor having multiple channels and multiple gates. The channels are connected in parallel between nodes A and N of Fig. 11(c). Switches (not shown in Fig. 11(c)) are coupled between the gates and the circuitry (not shown) providing the potential V_p , so that by closing each of the switches the gate coupled
- 20 thereto is held at potential V_p . The switches are controlled by dead zone width control bits $ctr[2:0]$ received from dead zone width control circuit 14. When circuit 14 increases the $ctr[2:0]$ value (to widen the dead zone in response to high jitter), more of the switches close, thus causing current to flow through more of the channels of the multi-finger transistor, and thereby increasing the absolute magnitude of I_p . When
- 25 circuit 14 decreases the $ctr[2:0]$ value (to narrow the dead zone in response to low jitter), more of the switches open, thus causing current to flow through fewer of the channels of the multi-finger transistor, and thereby decreasing the absolute magnitude of I_p . For example, in one implementation in which transistor P1 has eight channels, current flows through only one of the channels of transistor P1 when $ctr[2:0] = 000$,
- 30 and current flows through all of the channels of transistor P1 when $ctr[2:0] = 111$.

When NMOS transistor N1 of Fig. 11(c) is implemented as a multi-finger transistor having multiple gates and multiple channels, the channels of transistor N1 are connected in parallel between nodes N and B of Fig. 11(c). Switches (not shown in Fig. 11(c)) are coupled between the gates and the circuitry (not shown) providing the

potential V_n , so that by closing each of the switches the gate coupled thereto is held at potential V_n . The switches are controlled by dead zone width control bits $ctr[2:0]$ received from dead zone width control circuit 14. When circuit 14 increases the $ctr[2:0]$ value (to widen the dead zone in response to high jitter), more of the switches
 5 close, thus causing current to flow through more of the channels of the multi-finger transistor, and thereby increasing the absolute magnitude of I_P . When circuit 14 decreases the $ctr[2:0]$ value (to narrow the dead zone in response to low jitter), more of the switches open, thus causing current to flow through fewer of the channels of the multi-finger transistor, and thereby decreasing the absolute magnitude of I_P . For
 10 example, in one implementation in which transistor N1 has eight channels, current flows through only one of the channels of transistor N1 when $ctr[2:0] = 000$, and current flows through all of the channels of transistor N1 when $ctr[2:0] = 111$.

Preferably, multi-finger transistors P1 and N1, the switches (not shown) coupled to the gates of each of transistors P1 and N1, and the rest of the Fig. 11(c)
 15 operate to cause the absolute value of I_P to be proportional to the jitter, thereby causing the average pumping current (I_{avg}) sourced (or sunk) by circuitry 12 (over each time interval in which averaged phase error, between the raw data clock $idCLK$ and the data, is positive or negative), and data loop gain (K_{PD}), to be at least substantially independent of jitter and dead zone width (e.g., as shown in the bottom graph of Figure
 20 1(b), in which charge pump current (I_P) is varied to be proportional to dead zone width W_D to keep the data loop gain (K_{PD}) independent of jitter and dead zone width).

An important advantage of implementing charge pump circuitry 12 to include charge pump of Fig. 11(c) can be appreciated by comparing Figs. 11(a), 11(b), and 11(c). This advantage is the elimination or reduction of a charge sharing problem that
 25 occurs in many charge pumps (e.g., in the conventional charge pump shown in Fig. 11(a)). The parasitic capacitance at the $pint$ and $nint$ nodes in Figure 11(a) causes charge sharing, and can cause peak current to flow (to a capacitor being charged/discharged by the Fig. 11(a) circuit) if charge pump circuitry 12 of Fig. 3(a) is implemented to include the charge pump shown in Fig. 11(a). The simulation result
 30 shown in Fig. 11(a), which assumes that charge pump circuitry 12 is implemented to include the charge pump circuit of Fig. 11(a), shows about 300% of peak current flow in response to rapidly changing values of charge pump control bits “upb” and “down.” This peak current can undesirably cause a larger jitter and systematic phase offset in the PLL.

To overcome this problem, a charge pump with a charge removal transistor has been proposed in P. Larsson, "A 2-1600-MHz CMOS Clock Recovery PLL with Low-V_{dd} Capability," J. Solid-State Circuits, vol. 34, pp. 1951-1960, Dec. 1999. A charge pump having the Larsson design is shown in Fig. 11(b). Even using charge pumps
 5 having the Larsson design in charge pump circuitry 12 of Fig. 3(a), there still would be about 50% peak current flow (as shown in the simulation result of Figure 11(b), which assumes that charge pump circuitry 12 is implemented to include the charge pump circuit of Fig. 11(b)) in response to rapidly changing values of charge pump control bits "upb" and "downb."

10 In accordance with the present invention, such peak current flow is eliminated completely by inserting a delay element in the up/down signal path of each charge pump circuit of the inventive CDR. Figure 11(c) is a schematic diagram of a charge pump circuit including such a delay element (identified by reference numeral 50), and a graph (a simulation result) of the charge pump current generated by this charge pump in
 15 response to rapidly changing values of the charge pump control bits "up" and "downb" of Fig. 11(c). The simulation result of Fig. 11(c) shows that no peak current would flow to the capacitance being charged/discharged by the Fig. 11(c) circuit if each charge pump circuit of charge pump circuitry 12 were implemented as shown in Fig. 11(c).

AES PD 2 realizes the dead zone using 2x oversampling clocks. In contrast with
 20 a conventional 2x oversampling PD, each edge sampling clock $xCLK$ produced by AES PD 2 samples data with some offset from the data transition edge. By alternating the polarity of the offset, the dead zone is established. Each of circuits 2a, 2b, 2c, and 2d of AES PD 2 has two phase detection states, referred to as the UP state and the DOWN state, and the state of each circuit 2a, 2b, 2c, and 2d determines the polarity of sampling
 25 offset. Figure 3(c) shows clock positions for each state of circuit 2a, 2b, 2c, or 2d. As shown in Fig. 3(c), in the UP state, the edge sampling clock ($xCLK$) samples the data at the left side of the data transition edge (as shown in Figs. 1(b) and 4(a)), assuming the sampling clocks are locked to the data. If the "edge" data sample (the data sample most recently generated using the edge sampling clock of the relevant one of circuits 2a, 2b,
 30 2c, and 2d) does not match the most recently generated "center" data sample (where each "center" data sample is generated using one of the data sampling clocks $dCLK$), the up signal is asserted with a "one" value ($up = 1$) and the PD remains in the UP state (i.e., up_det remains equal to 1). If no transition edge is detected using such edge sampling clock (i.e., if the most recently generated "edge" data sample matches the

most recently generated “center” data sample), the *up* signal is asserted with a “zero” value and the PD undergoes a transition to the DOWN state (i.e., *up_det* = 0 is asserted).

As also apparent from Fig. 3(c) and the description below of Figs. 4(a) and 4(b),
 5 in the DOWN state, the edge sampling clock (*xCLK*) samples the data at the right side of the data transition edge (as shown in Figs. 1(b) and 4(a)), assuming the sampling clocks are locked to the data. If the “edge” data sample (the data sample most recently generated using the edge sampling clock of the relevant one of circuits 2a, 2b, 2c, and 2d) matches the next generated “center” data sample, the *down* signal is asserted with a
 10 “one” value (*down* = 1) and the PD remains in the DOWN state (i.e., *up_det* remains equal to 0). If a transition edge is detected using such edge sampling clock (i.e., if the most recently generated “edge” data sample does not match the next generated “center” data sample), the *down* signal is asserted with a “zero” value and the PD undergoes a transition to the UP state (i.e., *up_det* = 1 is asserted).

15 Figure 3(b) is a block diagram of an implementation of dead zone generator 10 of the Figure 3(a) circuit. The dead zone generator of Fig. 3(b) generates the variable offsets of the edge sampling clock *xCLK*. It modulates the raw edge sampling clock *ixCLK* according to the PD state (which is determined by the control bit *up_det*) and in response to dead zone width control bits (*ctr*[2:0]) from dead zone width control circuit
 20 14. A variable delay line (VDL 31) is used to establish the dead zone. By delaying the raw data sampling clock (*idCLK*) by half the length (half the total delay) of the VDL (to generate the data sampling clock *dCLK*), the edge sampling clock *xCLK* output from (asserted at the output of multiplexer 30 of) the Fig. 3(b) circuit in response to the raw edge sampling clock *ixCLK* can have a positive or negative offset relative to the data
 25 transition edge when the CDR is locked. VDL 31’s total delay is determined by the dead zone width control bits (*ctr*[2:0]), which are generated in a manner described with reference to Fig. 12(a). In response to increasing values of *ctr*[2:0], the width of the dead zone is increased. In response to decreasing values of *ctr*[2:0], the width of the dead zone is decreased.

30 Figure 4(a) shows the behavior of each of circuits 2a, 2b, 2c, and 2d of Fig. 3(a) in some representative cases. In contrast with the conventional binary PD, the inventive AES PD detects only one side of the phase error. In the UP state, the edge sampling clock *xCLK* (one of *xCLK*[0], *xCLK*[1], *xCLK*[2], and *xCLK*[3]) samples data at the left boundary of the dead zone and the AES PD makes a decision as to whether the

VCO phase should be increased or not, while the *down* signal is kept low. In the UP state (i.e., if $up_det = 1$), if a transition edge is detected using the edge sampling clock, the *up* signal is asserted ($up = 1$) and the AES PD remains in the UP state. If no transition edge is detected using the edge sampling clock, the *up* signal is not asserted
5 ($up = 0$) and the AES PD undergoes a transition to the DOWN state but the data loop does not change the raw edge sampling clock $ixCLK$ or the raw data sampling clock $idCLK$.

On the contrary, the *up* signal is always zero in DOWN state. In the DOWN state, the edge sampling clock $xCLK$ samples data at the right boundary of the dead
10 zone and the AES PD makes a decision as to whether the VCO phase should be decreased or not. In the DOWN state (i.e., if $up_det = 0$), if no transition edge is detected using the edge sampling clock, the *down* signal is asserted ($down = 1$) and the AES PD remains in the DOWN state. If a transition edge is detected using the edge sampling clock, the *down* signal is not asserted ($down = 0$) and the AES PD undergoes
15 a transition to the UP state but the data loop does not change the raw edge sampling clock $ixCLK$ or the raw data sampling clock $idCLK$.

More specifically, a preferred embodiment of finite state machine (FSM) 4 of each of circuits 2a, 2b, 2c, and 2d of Fig. 3(a) makes a decision as to whether to assert an *up* signal ($up[i]$, for $i = 0, 1, 2$, or 3) or a *down* signal ($down[i]$, for $i = 0, 1, 2$, or 3)
20 by comparing three data samples:

a first sample (to be denoted as a “P” sample, where “P” denotes previous or recent) of the data generated at an edge of the data sampling clock $dCLK[n]$;

a second sample (to be denoted as an “N” sample, where “N” denotes next) of the data generated at the next edge of the data sampling clock $dCLK[n+1]$, for $n = 0, 1$,
25 or 2 (in the case that $n = 3$, the “N” sample is generated at the next edge of data sampling clock $dCLK[0]$); and

a third sample (to be denoted as an “E” sample, where “E” denotes edge) of the data generated at an edge of the edge sampling clock $xCLK[n]$, said edge occurring between the edges of the data sampling clocks at which the “P” and “N” samples are
30 generated (e.g., between the edges of $dCLK[n]$ and $dCLK[n+1]$, when $n = 0, 1$, or 2).

FSM 4 of each of circuits 2a, 2b, 2c, and 2d of Fig. 3(a) preferably shares data samples with FSM 4 of another one of circuits 2a, 2b, 2c, and 2d, as follows. FSM 4 of circuit 2b employs the current “N” sample of circuit 2a as its “P” sample, FSM 4 of circuit 2c employs the current “N” sample of circuit 2b as its “P” sample, and so on.

In the UP state, each FSM 4 compares the P and E samples. If $P = E$, neither the *up* signal nor the *down* signal is asserted ($up = 0$ and $down = 0$), and the state is switched to the DOWN state. In the UP state, if P does not equal E, the *up* signal is asserted ($up = 1$ and $down = 0$), and the relevant one of circuits 2a, 2b, 2c, and 2d
 5 remains in the UP state.

In the DOWN state, each FSM 4 compares the N and E samples. If $N = E$, neither the *up* signal nor the *down* signal is asserted ($up = 0$ and $down = 0$), and the state is switched to the UP state. In the DOWN state, if N does not equal E, the *down* signal is asserted ($up = 0$ and $down = 1$), and the relevant one of circuits 2a, 2b, 2c, and
 10 2d remains in the DOWN state.

Figure 4(b) is a state transition diagram of each of circuits 2a, 2b, 2c, and 2d of Fig. 3(a). The next state of each circuit depends on a current state and a phase detection result (e.g., assertion of $up = 1$ when $up_det = 1$ in the UP state, or assertion of $down = 1$ when $up_det = 0$ in the DOWN state). If an $up = 1$ signal is detected in the UP state, the AES PD decides to stay in the UP state, and FSM 4 asserts $up_det = 1$. If a $down = 1$ signal is detected in the DOWN state, the AES PD decides to stay in the DOWN state, and finite state machine 4 asserts $up_det = 0$. The AES PD stays in the UP state until it fails to detect the up signal (i.e., until finite state machine 4 detects $up = 0$). The AES PD stays in the DOWN state until it fails to detect the down signal (i.e., until
 15 finite state machine 4 detects $down = 0$). Once the AES PD, in the UP (DOWN) state, fails to detect the up (down) signal, it changes its state to DOWN (UP) state in the next cycle. As a result, when the VCO clocks lag the input data, the AES PD mainly stays in the UP state, and when the VCO clocks lead the input data, the AES PD mainly stays in the DOWN state. Hence the AES PD can have a comparable phase acquisition speed
 20 with the conventional binary PD. When the CDR is locked to the input data, AES PD alternates its state at every data transition.

Fig. 12(a) is a block diagram of a dead zone width control loop that can be used to implement the dead zone width control loop of Fig. 3(a), with the bit " v_i " in Fig. 12(a) denoting one of the valid data transition bits $v[0:3]$ of Fig. 3(a). Dead zone width control circuit 14 of Fig. 3(a) can be implemented as the Fig. 12(a) circuit, to generate a set of dead zone width control bits (" $ctr[2:0]$ ") for use by circuits 10 (of AES PD 2) in generating the edge sampling clock signals $xCLK$.
 30

As noted, circuit 14 receives from AES PD 2 a sequence of bits up_i (from one of the layers of AES PD 2), a sequence of bits $down_i$ of the above-mentioned type (from

the same layer of AES PD 2), and a sequence of bits v_i (from the same layer of AES PD 2), each bit v_i indicating whether there has been a transition (during the last two cycles of clock $dCLK$) of the input data signal RDP, RDN.

The circuit of Fig. 12(a) counts (in the digital domain) the number of “ up_i ” and “ $down_i$ ” bits (received at the inputs of OR gate 91) that satisfy “ up_i ” = 1 or “ $down_i$ ” = 1 and that occur during a predetermined number, X, of valid data transitions (as indicated by X occurrences of $v_i = 1$ in the received sequence of bits v_i). The dead zone width control loop of the Fig. 3(a) system keeps the occurrence rate of $up_i = 1$ or $down_i = 1$ bits in a specific range (e.g., $1/16 \geq Y/X \geq 1/64$, where Y is the number of occurrences of $up_i = 1$ or $down_i = 1$ per X occurrences of $v_i = 1$ in a preferred embodiment) by counting the number of data transitions (each indicated by $v_i = 1$) and the number of $up_i = 1$ or $down_i = 1$ bits.

Counter 90 of Fig. 12(a) counts valid data transitions by counting occurrences of $v_i = 1$. During 2^{13} data transitions, occurrences of $up_i = 1$ or $down_i = 1$ are accumulated by counter 92, and the result is fed from counter 92 to decision logic 93. Logic 93 asserts one of five code words in response at the end of each cycle of 2^{13} data transitions: a “strong down” code word indicating a very small number of occurrences of $up_i = 1$ or $down_i = 1$ (e.g., from 000000000 to 000011111 occurrences) during the cycle; a “weak down” code word indicating a greater number of occurrences of $up_i = 1$ or $down_i = 1$ (e.g., from 000100000 to 000111111 occurrences) during the cycle; a “stay” code word indicating a greater number of occurrences of $up_i = 1$ or $down_i = 1$ (e.g., from 001000000 to 001111111 occurrences) during the cycle; a “weak up” code word indicating a greater number of occurrences of $up_i = 1$ or $down_i = 1$ (e.g., from 010000000 to 011111111 occurrences) during the cycle; and a “strong up” code word indicating a greater number of occurrences of $up_i = 1$ or $down_i = 1$ (e.g., from 100000000 to 111111111 occurrences) during the cycle.

The output of logic 93 is asserted to finite state machine (FSM) 94. FSM 94 includes logic that generates dead zone width control signals (P_{UP} , P_{DOWN}) and implements a hysteresis in generating the dead zone width control signals to stabilize the loop as indicated in the state transition diagram of Fig. 12(b). Figure 12(b) is a state transition diagram of FSM block 94 of Fig. 12(a). As indicated in Fig. 12(b), FSM 94 is in a “STAY” state in which it asserts bits P_{UP} and P_{DOWN} with the values $P_{UP} = 0$, $P_{DOWN} = 0$, unless and until it receives a “strong up” or “strong down” code word. In response to a “strong up” code word, it enters an “UP” state in which it asserts bits P_{UP} and

P_{DOWN} with the values $P_{UP} = 1$, $P_{DOWN} = 0$. In response to a “strong down” code word, it enters a “DOWN” state in which it asserts bits P_{UP} and P_{DOWN} with the values $P_{UP} = 0$, $P_{DOWN} = 1$.

In the “UP” state, FSM 94 enters the DOWN state in response to a “strong
5 down” code word, enters the STAY state in response to a “stay” or “weak down” code word, and otherwise remains in the UP state. In the “DOWN” state, FSM 94 enters the UP state in response to a “strong up” code word, enters the STAY state in response to a “stay” or “weak up” code word, and otherwise remains in the DOWN state.

Preferably, counter 95 initially asserts a dead zone width control word $ictr[2:0]$
10 (to one input of multiplexer 96) with the value $ictr[2:0] = 100$. In response to $P_{UP} = 1$, $P_{DOWN} = 0$, counter 95 increments by one the word $ictr[2:0]$ that it asserts to multiplexer 96. In response to $P_{UP} = 0$, $P_{DOWN} = 1$, counter 95 decrements by one the dead zone width control word $ictr[2:0]$ that it asserts to multiplexer 96. In response to $P_{UP} = 0$, $P_{DOWN} = 0$, counter 95 neither increments nor decrements the dead zone width
15 control word $ictr[2:0]$ that it asserts to multiplexer 96.

The default control word “111” is asserted to the other input of multiplexer 96. The control signal DzctrEN asserted to multiplexer 96 determines whether multiplexer 96 asserts the current dead zone width control word $ictr[2:0]$, or the default control word, as the current dead zone width control word $ctr[2:0]$. The control signal DzctrEN
20 is generated from the frequency lock signal produced by lock detection circuit 16, so that the dead zone width control loop is activated only after a frequency lock is achieved.

Circuits 90, 92, and 93 of Fig. 12(a) comprise a jitter estimating circuit. The “strong down” code word that it generates indicates a small amount of jitter in the input
25 data, the “weak down” code word indicates a greater amount of jitter, the “stay” code word indicates a greater amount of jitter, the “weak up” code word indicates a greater amount of jitter, and the “strong up” code word indicates a still greater amount of jitter. In response to each jitter signal generated by circuitry 14, the dead zone width control loop adjusts the width (W_D) of the dead zone determined by one of the edge sampling
30 clocks $xCLK$ (as described with reference to Fig. 3(b)), and preferably also controls the absolute magnitude of the charge pump current (I_P) sourced or sunk by charge pump circuitry 12 (as described with reference to Fig. 11(c)). The dead zone width (W_D) is automatically adjusted during operation (in response to the jitter signal, with hysteresis as implemented by finite state machine 94) to cause the edges of the dead zone to track

the boundaries of the data eye, for example as indicated in Figure 1(b). In Fig. 1(b), the data sampling clock $dCLK$ has phase error ϕ_A (relative to the center of the data eye), two versions of the edge sampling clock $xCLK$ are shown (an “up” version having phase $(\phi - 180 + W_D/2)$ degrees, and a “down” version having phase $(\phi - 180 - W_D/2)$ degrees, where ϕ is the phase of $dCLK$), and the dead zone width W_D is varied to be proportional to the data’s jitter.

A fully integrated test circuit for the inventive CDR can have the structure shown in Figure 5. In this test circuit, the CDR is a transceiver which includes transmitter circuitry 30 and receiver circuitry 31. Transmitter circuitry 30 serializes 16-bit parallel data into one pair of PECL level signals (TXN and TXP, indicative of a serial differential signal) via 16:4 MUX 32, serializer 40, and output driver 41, connected as shown. Preferably, a current mode driver adopting inductive load is used for bandwidth boosting, for example as described in M. M. Green, et al., “OC-192 Transmitter in Standard 0.18 μ m CMOS,” ISSCC Digest of Technical Papers, pp. 248-249, Feb. 2002. In receiver circuitry 31, serial input (RXN and RXP, indicative of a serial differential signal) is recovered as 16-bit parallel data through an embodiment of the inventive AES CDR (circuit 33) and 4:16 DEMUX 34, connected as shown. An LVDS interface can be used for I/O of the parallel data.

Transmitter 30 includes a phase lock loop circuit (TXPLL 38) for generating internal clock and sync signals in response to external reference clock (“Reference CLK”). AES CDR 33 also includes a phase lock loop for generating clock signals $ixCLK$ and $idCLK$ signals in response to the external reference clock.

To resolve an asynchronous clock boundary problem, elastic buffers 35 are used in the transmitter. For the convenience of measurement, built-in self test (“BIST”) circuitry (“BIST_RX” 36 in the receiver; “BIST_TX” 37 in the transmitter) is also included. The BIST circuitry supports jitter and BER testing for various data patterns including a pseudo-random bit stream (“PRBS”) data sequence. Multiplexer 43 selects the output of BIST_RX 36 in a test mode and otherwise selects the output of demultiplexer 34. Multiplexer 44 selects the output of BIST_TX 37 in a test mode and otherwise selects the output of buffers 35.

The test circuit of Fig. 5 has been fabricated as an integrated circuit with a 0.18 μ m CMOS process (a 0.18 μ m, 1-Poly, 4-Metal CMOS process) and packaged with

144-pin chip array BGA, with a pad-limited die size of 2.8×2.8 mm, and an active area of 0.5×1.5 mm².

The transceiver of the integrated circuit implementation of the Fig. 5 operates (during generation of samples at a rate in the range from 2.5 Gbps to 11.5 Gbps) with a
5 bit error rate (BER) of less than 10^{-12} . During a test, when the receiver of the chip was locked to 10 Gbps PRBS input data, it employed a sample clock having 2.5 ps RMS and 21 ps P-P jitter to recover serial data having 5.2 ps RMS and 34 ps P-P jitter. During a test, when the receiver of the chip was locked to 2.5 Gbps PRBS input data, it employed a sample clock having 3.9ps RMS and 38ps P-P jitter to recover serial data
10 having 7.6ps RMS and 52ps P-P jitter. The chip consumes 540 mW from a 1.8V supply when operating at 10 Gbps. The power dissipation is reduced to 340 mW when operating at 2.5Gbps.

We next describe additional aspects of the invention.

Because a binary PD has a non-linear characteristic, it is hard to apply a linear
15 PLL theory to a CDR in which the binary PD is used. But in a noisy environment, the input data has probabilistic distribution, and the inventors have recognized that this makes it possible to analyze the characteristic of the binary PD using random process theory.

Figure 6(a) shows an edge sampling clock of a conventional binary PD and the
20 input data transition probability density function (PDF) in a lock state, assuming no phase error (ϕ_d) between the sampling clock and data. Figure 6(b) shows an edge sampling clock of the conventional binary PD and the input data transition PDF, assuming nonzero phase error (ϕ_d) between the sampling clock and data. The expressions P(up) and P(down) denote the probability of an up decision (to advance the
25 phase of the sampling clock) and a down decision (to retard the phase of the sampling clock), respectively.

Under the conditions of Fig. 6(a), the binary PD is locked when $P(\text{up})$ is equal to $P(\text{down})$, and the average pumping current is equal to zero. But if there is some phase error between the sampling clock and the input data as shown in Figure 6(b), the average pumping current changes to

$$\begin{aligned} I_{avg} &= I_P \alpha_T (P(\text{up}) - P(\text{down})) \\ &= I_P \alpha_T \left(\int_{-\infty}^{\phi_\Delta} f(x) dx - \int_{\phi_\Delta}^{\infty} f(x) dx \right), \quad (1) \end{aligned}$$

where I_P , α_T , and $f(x)$ are a charge pump current, data transition rate, and data transition PDF, respectively. The phase detector gain (K_{PD}) is a slope of the $I_{avg}-\phi_\Delta$ curve, and can be expressed as follows.

$$\begin{aligned} K_{PD} &= \left. \frac{\partial I_{avg}}{\partial \phi_\Delta} \right|_{\phi_\Delta=0} = I_P \alpha_T \left[(f(\phi_\Delta) - f(-\infty)) - (f(\infty) - f(\phi_\Delta)) \right] \Big|_{\phi_\Delta=0} \\ &= 2 I_P \alpha_T f(0) \end{aligned} \quad (2)$$

Equation (2) implies that K_{PD} depends on the data transition PDF (i.e., the PDF of input jitter). Because $f(0)$ is inversely proportional to the jitter amplitude, the PD gain K_{PD} is also inversely proportional to the jitter. For example, if the input jitter is purely Gaussian, the resulting K_{PD} is

$$K_{PD} = \frac{2 \alpha_T}{\sigma \sqrt{2\pi}}, \quad (3)$$

where σ is the RMS value of the jitter.

A dead-zone binary PD based on 3x oversampling technique was proposed in the above-cited paper by Y. Moon, et al. Applying the same method used in the two previous paragraphs, a gain (K_{PD}) of a dead-zone PD can be described in closed form.

Figures 7(a) and 7(b) show an edge sampling clock and data transition PDF for a conventional dead-zone PD, assuming zero and nonzero phase error (ϕ_Δ), respectively. The average pumping current for phase error of ϕ_Δ is

$$\begin{aligned} I_{avg} &= I_P \alpha_T (P(\text{up}) - P(\text{down})) \\ &= I_P \alpha_T \left(\int_{-\infty}^{\phi_\Delta - W_D/2} f(x) dx - \int_{\phi_\Delta + W_D/2}^{\infty} f(x) dx \right), \end{aligned} \quad (4)$$

where W_D is a width of dead-zone. The PD gain can be found by differentiating equation (4):

$$\begin{aligned} K_{PD} &= \left. \frac{\partial I_{avg}}{\partial \phi_{\Delta}} \right|_{\phi_{\Delta}=0} = I_P \alpha_T [(f(\phi_{\Delta} - W_D/2) - f(-\infty)) - (f(\infty) - f(\phi_{\Delta} + W_D/2))] \Big|_{\phi_{\Delta}=0} \\ &= I_P \alpha_T (f(W_D/2) + f(-W_D/2)) \end{aligned} \quad (5)$$

- 5 As apparent from equation (5), the gain of dead-zone PD is also dependent on the jitter PDF. If the width of dead-zone is wide enough to locate its boundaries at tails of jitter PDF, the gain K_{PD} is inversely proportional to the jitter amplitude just as in a conventional binary PD.

The PD of the invention controls its width to track the jitter amount by locating
10 the dead-zone boundaries in jitter tails. A charge pump current, I_P is also controlled to be proportional to dead-zone width, so that gain K_{PD} can be kept constant in spite of jitter variations.

Figure 8 shows an edge sampling clock and data transition PDF for an
embodiment of the inventive PD, assuming nonzero phase error (ϕ_{Δ}). The PD gain is
15 already described in equation (5), but the PDF of jitter distribution is hard to know in the real world. Instead, an approximation of K_{PD} between $\phi_{\Delta}=0$ and $\phi_{\Delta}=W_D/2$ can be used to verify the inventive PD. When $\phi_{\Delta}=W_D/2$, the sampling clocks and the data edge distributions are located as shown in Figure 8. Then, the average pumping current is

$$\begin{aligned} I_{avg} &= I_P \alpha_T (P(up) - P(down)) \\ &\approx \frac{1}{2} I_P \alpha_T, \quad (\because P(up) = \frac{1}{2}, \quad P(up) \gg P(down)) \end{aligned} \quad (6)$$

20 So, the approximated PD gain is:

$$K_{PD}' = \frac{I_{avg}}{W_D/2} = \frac{I_P \alpha_T}{W_D}. \quad (7)$$

Because I_P is proportional to W_D in preferred embodiments of the inventive PD, K_{PD}'
25 is constant over various jitter conditions.

A ring oscillator type VCO is preferably used for multi-phase clocking when implementing the invention. To make the VCO have enough range to cover PVT (process, voltage, and temperature) variations while keeping a small gain for a low jitter, a dual control path scheme similar to that described in J. Savoj, et al., "A 10-Gb/s

CMOS Clock and Data Recovery Circuit with a Half-Rate Linear Phase Detector,” J. Solid-State Circuits, vol. 36, pp. 761-767, May 2001 and shown in Figures 9(a) and 9(b), is employed in preferred embodiments of the invention (including that to be described with reference to Figs. 9(a) and 9(b).

5 Fig. 9(a) is a block diagram of some elements of a transceiver that includes an embodiment of the inventive CDR, with simplified schematic diagrams of preferred implementations of VCOs 20. PLL 38 of Fig. 9(a) (identified as “TXPLL”) is included in the transmitter portion of the transceiver, and PLL 39 (identified as “RXPLL”) is included in the CDR portion of the transceiver. PLL 39 includes elements 2, 12, 13, 15,
10 19, and a preferred implementation of VCO 20 of Fig. 3(a), connected as shown.

With reference to Fig. 9(a), VCO 20 in TXPLL 38 is coupled (as shown) to receive different control voltages than is VCO 20 in RXPLL 39. A coarse control voltage (V_{coarse}) for RXPLL 39 is fed (to VCO 20 in RXPLL 39) from node M' of the low pass filter (LPF) of TXPLL 38. A fine control voltage (V_{fine}) for RXPLL 39 is fed
15 (to VCO 20 in RXPLL 39) from node M of low pass filter 15 of RXPLL 38. A single control voltage is fed (to VCO 20 of TXPLL 38) from node M of the low pass filter (LPF) of TXPLL 38.

Figure 9(b) is a schematic diagram of a single stage delay cell of VCO 20 of RXPLL 39 of Figure 9(a). The delay cell is of a folded starved inverter type for low
20 jitter. In response to the clock signals “in” and its complement “inb,” the cell generates the output clock signals “out” and its complement “outb.”

The inventive CDR can operate at a frequency equal to one quarter of the sample rate (i.e., the combined rate at which the samples of sample streams DS[0:3] are asserted can be four times the frequency of each of clocks $ixCLK$ and $idCLK$). Since
25 AES PD 2 of Fig. 3(a) modulates the edge sampling clock, the CDR of Fig. 3(a) preferably operates at 2.5GHz for a 10 Gbps data rate. To minimize the probability of a metastable state, each sampler (e.g., sampler circuits 6 and 8 of Fig. 3(a)) preferably includes cascaded sense amps (51 and 52), a dynamic latch (53), and an inversion and delay circuit (54), connected as shown in Figure 10. First stage sampling amp 51
30 operates at the falling edge of CKS (i.e., $xCLK$ or $dCLK$ of Fig. 3(a)), while second stage sampling amp 52 operates at the rising edge of CLK . Inverter delay circuit 54 in the sampling clock path ensures stable pre-charge and evaluation operation of each sampling amp stages.

Another aspect of the invention is a method for sampling of data having jitter, including the steps of: (a) generating at least one sampling clock in response to a charge pump current, where at least one said sampling clock is a data sampling clock and the charge pump current determines the phase of the data sampling clock; (b) generating
5 data samples by sampling the data using the data sampling clock; and (c) generating the charge pump current in response to feedback, where the feedback is indicative of phase error between the data sampling clock and the data, the feedback is also indicative of the jitter, the charge pump current has an average current value that is at least substantially independent of the amount of the jitter over each time interval over which
10 ϕ_{av} is nonzero, where ϕ_{av} is an average of instantaneous values of said phase error over the time interval, and the average current value is an average of instantaneous values of the charge pump current over said time interval. Typically, the charge pump current is a positive current (I_P) when the phase error is negative, the charge pump current is a negative current, $-(I_P)$, when the phase error is positive, and step (c) includes the step of
15 controlling the absolute value of the charge pump current to cause said absolute value be proportional to the jitter. Preferably, another sampling clock generated during step (a) is a first clock whose phase is determined by the charge pump current, and step (c) includes the steps of: modulating the first clock to generate an edge sampling clock such that the edge sampling clock defines a dead zone having a dead zone width;
20 generating additional samples of the data in response to the edge sampling clock; and generating the feedback in response to the data samples and the additional data samples. In some embodiments, the data determines a data eye, and step (c) includes the step of automatically adjusting the dead zone width to cause edges of the dead zone to track boundaries of the data eye.

25 It should be understood that while some embodiments of the present invention are illustrated and described herein, the invention is defined by the claims and is not to be limited to the specific embodiments described and shown.